

# STA 445 S24 Assignment 5

Gabriel Cage-Sepeda

2024-03-05

```
library(tidyverse)
library(stringr)
```

## Problem 1

For the following regular expression, explain in words what it matches on. Then add test strings to demonstrate that it in fact does match on the pattern you claim it does. Do at least 4 tests. Make sure that your test set of strings has several examples that match as well as several that do not. Make sure to remove the `eval=FALSE` from the R-chunk options.

- a. This regular expression matches: *must contain lowercase a*

```
strings <- c("ready", "man", "nope", "And")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, 'a') )
```

```
##   string result
## 1 ready    TRUE
## 2  man    TRUE
## 3 nope   FALSE
## 4  And   FALSE
```

- b. This regular expression matches: *must contain lowercase a immediately followed by lowercase b*

```
strings <- c("absolute", "bank", "ready", "best", "ABSOLUTE", "BANK")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, 'ab') )
```

```
##   string result
## 1 absolute  TRUE
## 2  bank   FALSE
## 3 ready   FALSE
## 4  best   FALSE
## 5 ABSOLUTE FALSE
## 6  BANK   FALSE
```

- c. This regular expression matches: *must contain a or b or both, must be lower case, can be in any order*

```
strings <- c("absolute", "bank", "ABSOLUTE", "BANK", "ready", "best", "axxb", "bxxa")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '[ab]') )
```

```
##      string result
## 1 absolute   TRUE
## 2    bank    TRUE
## 3 ABSOLUTE FALSE
## 4    BANK   FALSE
## 5   ready   TRUE
## 6    best   TRUE
## 7   axxb   TRUE
## 8   bxxa   TRUE
```

- d. This regular expression matches: *must contain a or b or both, must be lowercase and must be at the beginning of the word*

```
strings <- c("absolute", "bank", "ABSOLUTE", "BANK", "ready", "best", "xxabxx", "ax")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '^[ab]') )
```

```
##      string result
## 1 absolute   TRUE
## 2    bank    TRUE
## 3 ABSOLUTE FALSE
## 4    BANK   FALSE
## 5   ready   FALSE
## 6    best   TRUE
## 7  xxabxx   FALSE
## 8     ax    TRUE
```

- e. This regular expression matches: *must contain at least one number and at least one 'a', can be uppercase or lowercase, the number must come before the a and there must be only exactly one space between the number and a.*

```
strings <- c("1 a", "1a", "1 1 a", "111 A", "1 a", "a 1", "1 aa", "a1m ba3", "m1m bab", "m1 ab")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '\\d+\\s[aA]') )
```

```
##      string result
## 1    1 a   TRUE
## 2    1a  FALSE
## 3   1 1 a   TRUE
## 4  111 A   TRUE
## 5    1 a  FALSE
## 6    a 1  FALSE
## 7   1 aa   TRUE
## 8 a1m ba3 FALSE
## 9 m1m bab FALSE
## 10 m1 ab   TRUE
```

- f. This regular expression matches: *must contain at least one number and one ‘a’, uppercase or lowercase, the number must come before the a, and only white space or no space may exist between the number and a.*

```
strings <- c("1 a", "1a", "1 1 a", "111 A", "1 a", "a 1", "1 aa", "a1m ba3", "m1m bab", "m1 ab")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '\\d+\\s*[aA]') )
```

- g. This regular expression matches: *0 or more repetitions of any character; literally anything, even nothing works*

```
strings <- c(" ", "a", "1", "1a", "aaaaa", " ", "#", "")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '.*') )
```

- h. This regular expression matches: *must contain two ‘w’ immediately followed by ‘bar’ at the beginning of the string*

```
strings <- c("wwbar", " wwbar", "ww", "barww", "wwb", "wwabcbar", "ww bar", "wwbar1")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '^\\w{2}bar') )
```

- i. This regular expression matches: *must contain “foo.bar” exactly anywhere in the string OR “wwbar” exactly at the beginning of the string*

```
strings <- c("foo.bar", "wwbar", "foo bar", "bar", "xxfoo.barxx", "wwbarxx")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '(foo\\.bar)|(^\\w{2}bar)') )
```

## Problem 2

The following file names were used in a camera trap study. The S number represents the site, P is the plot within a site, C is the camera number within the plot, the first string of numbers is the YearMonthDay and the second string of numbers is the HourMinuteSecond.

```
file.names <- c( 'S123.P2.C10_20120621_213422.jpg',
                 'S10.P1.C1_20120622_050148.jpg',
                 'S187.P2.C2_20120702_023501.jpg')
```

Produce a data frame with columns corresponding to the site, plot, camera, year, month, day, hour, minute, and second for these three file names. So we want to produce code that will create the data frame:

Site	Plot	Camera	Year	Month	Day	Hour	Minute	Second
S123	P2	C10	2012	06	21	21	34	22
S10	P1	C1	2012	06	22	05	01	48
S187	P2	C2	2012	07	02	02	35	01

```
df2 <- data.frame(string = file.names)
separate(df2, string, sep = '_|\\.', into = c('Site', 'Plot', 'Camera', 'Dates', 'Times', 'jpg')) %>%
  mutate(Year = str_sub(Dates, start=1, end=4),
         Month = str_sub(Dates, start=5, end=6),
         Day = str_sub(Dates, start=7, end=8),
         Hour = str_sub(Times, start=1, end=2),
         Minute = str_sub(Times, start=3, end=4),
         Second = str_sub(Times, start=5, end=6)) %>%
  select(!c(Dates, Times, jpg))
```

```
##   Site Plot Camera Year Month Day Hour Minute Second
## 1 S123   P2    C10 2012    06  21   21    34    22
## 2  S10   P1     C1 2012    06  22   05     1    48
## 3 S187   P2     C2 2012    07  02   02    35     1
```

3. The full text from Lincoln's Gettysburg Address is given below. Calculate the mean word length *Note: consider 'battle-field' as one word with 11 letters*).

```
Gettysburg <- 'Four score and seven years ago our fathers brought forth on this
continent, a new nation, conceived in Liberty, and dedicated to the proposition
that all men are created equal. Now we are engaged in a great civil war, testing
whether that nation, or any nation so conceived and so dedicated, can long
endure. We are met on a great battle-field of that war. We have come to dedicate
a portion of that field, as a final resting place for those who here gave their
lives that that nation might live. It is altogether fitting and proper that we
should do this. But, in a larger sense, we can not dedicate -- we can not
consecrate -- we can not hallow -- this ground. The brave men, living and dead,
who struggled here, have consecrated it, far above our poor power to add or
detract. The world will little note, nor long remember what we say here, but it
can never forget what they did here. It is for us the living, rather, to be
dedicated here to the unfinished work which they who fought here have thus far
so nobly advanced. It is rather for us to be here dedicated to the great task
remaining before us -- that from these honored dead we take increased devotion
to that cause for which they gave the last full measure of devotion -- that we
here highly resolve that these dead shall not have died in vain -- that this
nation, under God, shall have a new birth of freedom -- and that government of
the people, by the people, for the people, shall not perish from the earth.'
```

```
Gettysburg <- str_replace_all(Gettysburg, pattern="\\--", replacement=" ")
Gettysburg <- str_replace_all(Gettysburg, pattern="\\. ", replacement="")
Gettysburg <- str_replace_all(Gettysburg, pattern="\\-", replacement="")
str_split(Gettysburg, '\\s+|\\n|\\r|\\s+')[[1]] %>%
  str_length() %>%
  mean()
```

```
## [1] 4.239852
```